

On the CDF Fast Navigator page, go to CVS browser. In there, you'll see that there are actually several things that look like they could be the HWW group's "Diboson" framework. Choose this one:

```
cvs co Diboson_v17
cd Diboson_v17
```

The last version of this framework was actually v21, but for some reason they just kept adding the new code to folders within this release and there are actually different versions of the numbered versions. You can see that in the folders here. Go to:

```
cd Diboson_v21
```

If you want to do anything with this framework, it will be here. This framework generates ntuples in two stages:

1. First, Stntuples are accessed to make object-oriented "Dbntuples"--in HWW group parlance. This is where a lot of the object identification and basic filtering happens. That code is in the `Diboson/` directory.
2. Second, from those "Dbntuples" a second set of flat ntuples called "ntps" are created. These are more usable for analysis purposes because they contain all the fully processed and corrected kinematic values and things like that. The code for this is in the `Ana/` directory.

If you need to actually go into the code of the `Diboson/` and `Ana/` directories, we can go into that in more detail, but I'll skip by that for now. The "ntps" that you would likely use are already available.

Let's compile this. To compile for the first time, use the script:

```
./Compile.sh
```

That will take a while.

I put a copy of all the v21 ntps I have in a folder on fcdflnx3 here:

```
/cdf/spool/jnett80/ntps_HWW/
```

So you should be able to access those.

Next, also copy over the `SelectEvents/` directory that is in the same location--I tried to remove all extraneous code. This contains the code I used to strip events from the ntps and a module I wrote to output tons of plots. In my code, I'm actually stripping out 3-lepton events in addition to some other cuts since that was my place in HWW. Going back through this, making a selection from the ntps is actually not easy. The code uses a cut mask approach to mix and match desired selection requirements. If you're not familiar with that, we can schedule a time to talk through it, but I'll do what I can here.

As it is, there is a module at `SelectEvents/src/SelectEventsFinal.cc`, so do:

```
make clean
```

```
make
```

from the `SelectEvents/` directory. The `Makefile` should already be setup. Open the file `src/SelectEventsFinal.cc` and find the line:

```
TString const BaseInDir = "/cdf/spool/jnet80/ntp_HWW/";
```

Change that directory to point to wherever you put your copy of the ntp files. Immediately following that line, two dynamic standard vectors are declared, called `FileNames` and `ProcNames`. Immediately following that are a bunch of commented-out lines that can add ntp file names to the vector--uncomment the filename that you want to run. Also, for each file name in `FileNames`, `ProcNames` (for "process names") must have an addition in the same order. You can see that those follow. By default, I have only the data ntp file uncommented. Let's try a test run with the code as-is:

```
source ./run.csh
```

I get this output:

```
-----
<fcdflnx3.fnal.gov> source ./run.csh
NEvents set to 2147483647
NLep set to 3
REGION set to BaseWHTrilep
StripFileName (root) set to
HistogramFiles_Final/HistFiles/nlep3_NoZPeakSignal_
Number of files: 1
    strip file name =
HistogramFiles_Final/HistFiles/nlep3_NoZPeakSignal_data.root
FILE: /cdf/spool/jnet80/ntp_HWW/ntp_Data_nodupl.root  Proc: data
InFileName: /cdf/spool/jnet80/ntp_HWW/ntp_Data_nodupl.root
INFO: Detected data file.  Running with no weight scaling
Stripping file set to:
HistogramFiles_Final/HistFiles/nlep3_NoZPeakSignal_data.root
    Nevents: 1420582
    GetEntries: 1.23457e+09
NEvents requested is larger then InChain.GetEntriesFast().  Using
Max.
Xsect =      0
BrFrac =      0
FilterEff = 0
Fraction = 1
cutMask:  000000000110000000000000100001111
invertWord:  000000000000000000000000100000000
realMask:  0000000001100000000000000000001111
Processing event: 0
Processing event: 100000
Processing event: 200000
Processing event: 300000
Processing event: 400000
Processing event: 500000
Processing event: 600000
```

```
Processing event: 700000
Processing event: 800000
Processing event: 900000
Processing event: 1000000
Processing event: 1100000
Processing event: 1200000
Processing event: 1300000
Processing event: 1400000
    nevents: 1420582
INFO: Making new branch for Integral
DeanIntegralSF: 1
WeightSum:          38.0000
WeightSumByNJet[2+]:          8.0000
FINISH
WRITE AND CLOSE
```

-----  
That produced a new file here:

```
<fcdflnx3.fnal.gov> ls HistogramFiles_Final/HistFiles/
nlep3_NoZPeakSignal_data.root
```

To make a different set of cuts, you'll have to hack the `cutMask` definition in `SelectEventsFinal.cc`. See line 1084 for a list of the available cuts. Notice in the printout about, I have `cutMask`, `invertWord`, and `realMask` printed. Each binary unit corresponds to a cut in that list (make the cut, true or false), from right to left. This will take some playing around with to get accustomed to.

Go to `HistogramFiles_Final/` and compile with "make". This will compile `scripts/StackValues.cc`, which is run with `src/RunStacksAndPlots.csh`. Looking at the run script, you can see that you'll have to make files for other processes than just data to use this, so feel free to use this script as a starting point for something else or ignore everything here entirely.